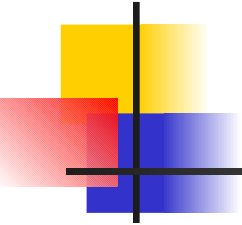


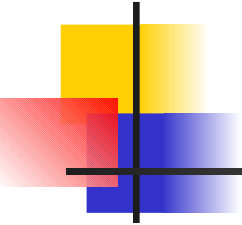
Iteration Planning

1. List the high level tasks derived from that previous iteration that your group will work on.
2. Refine tasks from the previous iterations into lower level tasks to be implemented.
3. Determine core functionalities, code dependencies, and bottlenecks among those tasks. Prioritize them based on that determination and the needs of the client.
4. Create a task chart showing what portion of each tasks has been completed over the given iteration. Display that chart as first slide of each iteration report along with the chart from the previous iteration.



Handling Code Dependencies.

1. Where task B depends on the code for task A, if possible one should not wait for the task A to be completed before starting to code task B. Usually it is possible to determine the output to be produced by the task A, and to construct a number of example data sets that can be used to test the development of the task B. Any function calls to task A can be simulate by stubs that deliver the example data sets.
2. Handling code dependencies requires that the team develop the data formats for the project very early in the planning process.



Unit Tests

1. When code is develop it must be tested. The tests designed debug the code should be cataloged and converted into unit tests.
2. These test must be associated with the code, perhaps as comments, in the svn. A piece of code cannot be consider complete without an attached list of unit tests.